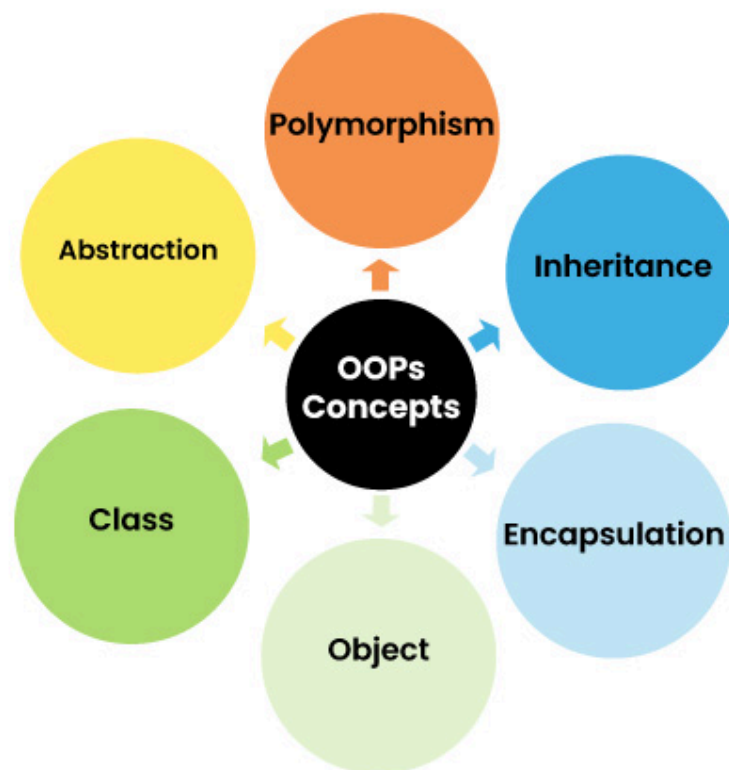


# Java - OOPs (Object-Oriented Programming) Concepts

## OOPs (Object-Oriented Programming System)

Object means a real-world entity such as a mobile, book, table, computer, watch, etc. Object-Oriented Programming is a methodology or paradigm to design a program using classes and objects. It simplifies software development and maintenance by providing some concepts.

In this tutorial, we will learn about the concepts of Java (OOPs) object-oriented programming systems.



## Java OOPs (Object-Oriented Programming) Concepts

- Object
- Class
- Inheritance
- Polymorphism

## 1. Object

In object-oriented programming, an object is an entity that has two characteristics (states and behavior). Some of the real-world objects are book, mobile, table, computer, etc. An object is a **variable** of the type class, it is a basic component of an object-oriented programming system. A class has the **methods** and **data members** (attributes), these methods and data members are accessed through an object. Thus, an object is an instance of a class.

## 2. Class

In object-oriented programming, a class is a blueprint from which individual objects are created (or, we can say a class is a **data type** of an object type). In Java, everything is related to classes and objects. Each class has its methods and attributes that can be accessed and manipulated through the objects.

## 3. Inheritance

In object-oriented programming, inheritance is a process by which we can reuse the functionalities of existing classes to new classes. In the concept of inheritance, there are two terms base (parent) class and derived (child) class. When a class is inherited from another class (base class), it (derived class) obtains all the properties and behaviors of the base class.

## 4. Polymorphism

The term "polymorphism" means "many forms". In object-oriented programming, polymorphism is useful when you want to create multiple forms with the same name of a single entity. To implement polymorphism in Java, we use two concepts **method overloading** and **method overriding**.

The method overloading is performed in the same class where we have multiple methods with the same name but different parameters, whereas, the method overriding is performed by using the inheritance where we can have multiple methods with the same name in parent and child classes.

## 5. Abstraction

In object-oriented programming, an abstraction is a technique of hiding internal details and showing functionalities. The abstract classes and interfaces are used to achieve abstraction in Java.

The real-world example of an abstraction is a Car, the internal details such as the engine, process of starting a car, process of shifting gears, etc. are hidden from the user, and features such as the start button, gears, display, break, etc are given to the user. When we perform any action on these features, the internal process works.



In an object-oriented approach, encapsulation is a process of binding the data members (attributes) and methods together. The encapsulation restricts direct access to important data. The best example of the encapsulation concept is making a class where the data members are private and methods are public to access through an object. In this case, only methods can access those private data.

Learn **Java** in-depth with real-world projects through our **Java certification course**. Enroll and become a certified expert to boost your career.

## Advantages of Java OOPs

The following are the advantages of using the OOPs in Java:

- The implementations of OOPs concepts are easier.
- The execution of the OOPs is faster than procedural-oriented programming.
- OOPs provide code reusability so that a programmer can reuse an existing code.
- OOPs help us to keep the important data hidden.